PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 7/10, 7/08 | A1

(11) International Publication Number: **WO 00/28739**

(43) International Publication Date: 18 May 2000 (18.05.00)

(21) International Application Number: PCT/US99/25730

(22) International Filing Date: 1 November 1999 (01.11.99)

(30) Priority Data:
09/189,733 10 November 1998 (10.11.98) US

(71) Applicant: OPEN TV, INC. [US/US]; 401 East Middlefield Road, Mountain View, CA 94043 (US).

(72) Inventor: FREIMANN, Felix; 718 Old San Francisco Road, #279, Sunnyvale, CA 94086 (US).

(74) Agents: YI, Susan, C. et al.; Ritter, Van Pelt & Yi LLP, Suite 205, 4906 El Camino Real, Los Altos, CA 94022 (US).

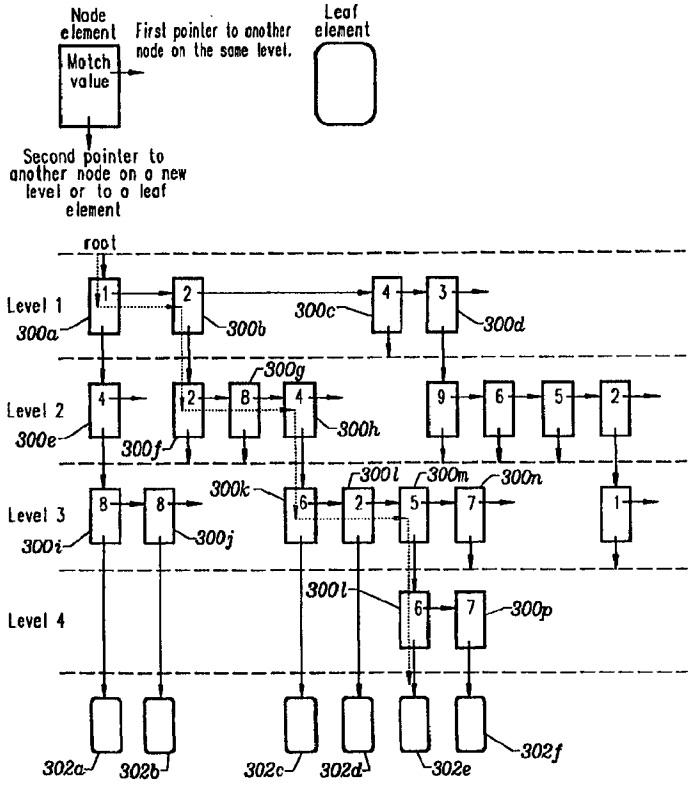
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: SYSTEM AND METHOD FOR INFORMATION FILTERING

(57) **Abstract**

According to an embodiment of the invention, a fast matching algorithm uses a memory space efficient data structure to accomplish information filtering in a system receiving information, such as a set top box used in conjunction with a television set. The data structure, according to an embodiment of the present invention, may include a node (300a) with a plurality of levels of expected values of data fields with an associated index to another node (300b) to be used when a matching identification of a particular level is not found. Additionally, according to an embodiment of the invention, the nodes (300a–p) in this data structure may include a level field which identifies levels which are valid within the node.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

SYSTEM AND METHOD FOR INFORMATION FILTERING

FIELD OF THE INVENTION

The present invention relates to electronic receipt of information. In particular, the present invention relates to software filtering of information received by a set top box.

BACKGROUND OF THE INVENTION

In a data receiving system, such as an interactive television system, information, such as service information (SI), may be sent from a head end in various blocks of data, sometimes referred to as sections, via various transport mediums. For example, data sections may be sent from a head end via cable or satellite to a set top box coupled to a television set. An example of the contents of such data sections include information regarding present or following events, services which the user of the set top box may access, and the current time and date. Examples of events may include a television program, or a commercial, while an example of services may be a set of channels subscribed to by a viewer.

A particular set top box need only acquire sections which are required by that particular set top box, sections which are new, or sections which have changed. For example, if a viewer has only subscribed to a set of basic channels, then the sections specific to a special sports channel or movie channel not included in the basic channel set would not be required by the set top box. Accordingly, all sections which do not meet a particular set top boxes criteria's may be

disregarded. The acceptance of a few select sections and rejection of all other sections may be accomplished though a combination of hardware and software filtering mechanisms.

A hardware filtering mechanism will typically perform a first round of filtering and pass on those sections which are accepted to a software filtering mechanism. The software filtering mechanism typically analyzes a header of the section, which is commonly the first 8 to 12 bytes of a section. When the start of a new section is detected, the section header is typically acquired form the transport stream. This section header is then passed to the software filter process which typically decides whether the remainder of the section should be acquired or rejected. If it is determined that the section is required, then the remainder of the section is acquired and passed on for further processing. Accordingly, only sections which have passed the software filter process will typically be forwarded for further processing.

A section may be uniquely identified by a number of individual fields within the section header. For example, depending of the section type, up to four individual fields may be used to identify a section. When a specific section is required, the software filter process may store the expected values of the individual data fields of the section header. When a section header is passed to the software filter, these expected values may be compared with the actual values from the section header. If no match occurs the next set of expected values may be compared until either a match occurs or no more expected values are available for comparison. Because the software filter may search a substantially large number of expected section header values, it is desirable to have a highly efficient comparison algorithm. A highly efficient comparison algorithm is particularly desirable since a problem may occur if a new section arrives before the software filter has finished evaluating a current section. This situation may cause an overflow or the new section may be lost. Additionally, a set top box may search for a variety of different sections at any one time, thus increasing the need for efficiency.

In addition to the need for efficiency, another common factor that is the need for memory. The memory usage for storage of the large number of expected values is typically a substantially large percentage of the very limited amount of memory commonly available in set top boxes.

It would be desirable to have a time efficient matching algorithm which uses a memory space efficient data structure. The present invention addresses such needs.

SUMMARY OF THE INVENTION

According to an embodiment of the present invention, a fast matching algorithm uses a memory space efficient data structure to accomplish information filtering in a system receiving information, such as a set top box used in conjunction with a television set. The data structure, according to an embodiment of the present invention, may include a node with a plurality of levels of expected values of data fields with an associated index to another node to be used when a matching identification of a particular level is not found. Additionally, according to an embodiment of the present invention, the nodes in this data structure may include a level field which identifies levels which are valid within the node.

In addition, according to an embodiment of the present invention, a section map may be used which utilizes a series of section maps, each section map including a predetermined number of bits, such as 32 bits. Each section map may include a single bit which indicates what is stored in the first 15 bits. For example, the last 16 bits of a section map may indicate whether the first 16 sections have been received. Another bit may be used to indicate what is stored in the first 15 bits. For example, the first 15 bits may indicate whether the next 15 sections have been received, or they may indicate an index pointing to the next section map.

A method according to an embodiment of the present invention for filtering information in an interactive television environment is presented. The method comprises providing a data

structure, wherein the data structure includes a plurality of levels; and comparing information with a first criteria, wherein the first criteria is related to a first one of the plurality of levels. The method also compares the information with a second criteria, wherein the second criteria is related to a second one of the plurality of levels, this comparison occurring if the information matches the first criteria.

A system according to an embodiment of the present invention for filtering information in an interactive television environment is also presented. The system comprises a memory configured to provide a data structure, wherein the data structure includes a plurality of levels. The system also includes a processor coupled to the memory, the processor being configured to compare the information with a first criteria, wherein the first criteria is related to a first one of the plurality of levels. The processor is also configured to compare the information with a second criteria, wherein the second criteria is related to a second one of the plurality of levels, this comparison occurring if the information matches the first criteria.

A method according to an embodiment of the present invention for recording receipt of information is also presented. The method comprises providing a section number, and providing a first section map, wherein the first section map is associated with a number of bits. The method also determines whether the section number is less than the number of bits; and points to a second section map if the section number is not less than the number of bits.

A system according to an embodiment of the present invention for recording receipt of information is also presented. The system comprises a memory configured to provide a first section map, the first section map being associated with a number of bits, and a processor coupled to the memory. The processor is configured to provide a section number; determine whether the section number is less than the number of bits; and point to a second section map if the section number is not less than the number of bits.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a system suitable for implementing an embodiment of the present invention.

Figure 2 is a block diagram of an example of a set top box suitable for implementing an embodiment of the present invention.

Figure 3 is an illustration of a parse tree which may be used for a software filter.

Figure 4 is an illustration of a section mapping which may also be used with a software filter.

Figure 5 illustrates a sub-table node data structure for use with a software filter according to an embodiment of the present invention.

Figure 6 is an illustration of relationships between sub-table nodes according to an embodiment of the present invention.

Figures 7 – 8 are flow diagrams of a method according to an embodiment of the present invention for filtering information sections.

Figures 9A – 9B are illustrations of a section map according to an embodiment of the present invention.

Figure 10 is a flow diagram according to an embodiment of the present invention for determining whether a section has already been received.

Figure 11 is a flow diagram according to an embodiment of the present invention for creating a new section map.

Figures 12A – 12D are illustrations of examples of a section header suitable for use with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following description is presented to enable one of ordinary skill in the art to make and to use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiments will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

Figure 1 is a block diagram of an information receiving system suitable for implementing an embodiment of the present invention. The system of Figure 1 is shown to include a head end 102 which may be coupled with a video and audio device 100 which feeds a particular video with associated audio to the head end 102. The video information may be digitized at the head end 102 and transmitted via a transmitter 104 to a receiving system. The information transmitted by the head end 102 may be transmitted to the receiving system in various ways. For example, the transmitted information may be sent to the receiving system via a satellite 106. Alternatively, a

cable may be used to transmit information from the head end 102 to a receiving system. An example of a receiving system includes a television 112 coupled with a set top box 110. The set top box 110 may be coupled with a receiving antenna 108 for receiving information from the satellite 106.

Once information is received through the receiving antenna 108, it may be processed by the set top box 110 and displayed on the television set 112. In this manner, audio, video, and interactive data may be received and processed by the set top box 110.

Figure 2 is a block diagram of a set top box, such as set top box 110 of Figure 1, suitable for implementing an embodiment of the present invention. The set top box 110 shown in Figure 2 is shown to be coupled with the receiving antenna 108 and the television 112. A signal received by the receiving antenna 108 is sent to a front end 200 which converts the modulated signals received from the receiving antenna 108 to a demodulated digital signal. The signals are then sent form the front end 200 to a micro-controller 202. The micro-controller may include a hardware filter, such as a private elementary stream (PES) and service information (SI) filter's 204. Although a single PES filter 204 is shown in Figure 2, a set of a plurality of PES filters may actually be used. The PES filter 204 may select packets out of the packet stream of the received signal. Once packets are selected by the PES filter 204, they may be sent to be processed by a software filter in the micro-controller 202. The software filter then further selects packets from the set of packets received from the PES filter 204. Digital video signals and digital audio signals are then sent to a digital-to-analog converter 210. There may be a digital-to-analog conversion for video as well as for audio.

The micro-controller 202 is also shown to be coupled with a FLASH memory 206 and a system random access memory (RAM) 208. Programs which are run by the micro-controller 202 may be located in the FLASH memory 206, while other information, such as a database, may be located in the system RAM 208.

The digital-to-analog converter 210 is also shown to be coupled with a memory for decompression 214 which may be utilized in decompressing compressed signals. The digital-to-analog converter 210 is also shown to be coupled with a remodulator 212 which can remodulate audio and video signals for proper format to be displayed on the television 112.

Data received by the set top box 110 may be intended to be formatted into various tables. When a series of packets are received, they may be formatted into tables such as a network information table (NIT), bouquet association table (BAT), service descriptor table (SDT), and event information table (EIT). These tables are described in the Digital Video Broadcasting (DVB) standard. For further information regarding the DVB standard, the following internet site may be referenced: http://www.atsc.org.

As the data stream is received, the PES filter, such as the PES filter 204 of Figure 2, performs an initial filter function of the packet stream. Thereafter, the packets which have been selected by the PES filter are then processed through a software filter. A micro-controller may include multiple PES filters, such as 16 to 32 hardware PES filters to process through packets which are coming into the system at a very fast rate. The incoming packet stream may be transmitted at 18 to 50 mega bits per second.

A receiving system may receive all types of tables, including NIT, BAT, SDT, and EIT. A television network may have a large number of services. The NIT identifies where these service may be found; the BAT may group these services into categories, such as movies or sports; the SDT indicates more detailed information on these services; and the EIT commonly indicates additional information regarding services, such as the title and language.

During the software filter process, the various types of tables may have different levels of filtering criteria. For example, an NIT table may have a level one filtering criteria of identifying a table ID, and a level two filtering criteria of a network ID. The table ID identified the table that a section will be a part of when it is formatted, such as NIT, BAT, etc. The network ID identified

the network. In this example, a BAT table may have a level one criteria of a table ID and a level two criteria of a bouquet ID. A bouquet may be a predetermined combination of services, such as a combination of sports programs and movie programs, that a subscriber may subscribe to. A bouquet ID identifies a particular type of bouquet. Also in this example, the SDT may have a level one criteria of a table ID, a level two criteria of a transport stream ID, and a level three criteria of an original network ID. A transport stream ID identifies which frequency a satellite is using in transmitting a signal. The original network ID identifies a network which originally broadcasts a particular service. An EIT might have a level one criteria of a table ID, a level two criteria of a service ID, a level three criteria of a transport stream ID, and a level four criteria of an original network ID. A service ID identifies a service within a specific transport stream ID.

A section is a transmission unit which includes a number of packets. The total number of packets included in a section may vary from table to table. For example, the largest section for NIT is approximately 1024 bytes. A table, such as an NIT, may be split into a plurality of sections, such as 256 sections. A section may include a header. An example of a number of bytes to be included in a section header is the first 8 to 12 bytes. When the start of a new section is detected, the section header may be acquired from the transport screen and passed to the software filter process to determine if the remainder of the section should be acquired. NIT, BAT and SDT sections may have up to 1024 Bytes length. However, the EIT section may have a length of up to 4096 Bytes.

As data packets are passed along from the hardware PES filters to the software filter, the software filter may then provide further filtering of the packet stream. One example of a conventional filtering process is illustrated in Figure 3. Figure 3 shows a binary tree including nodes 300A – 300P and leaves 302A – 302F. In the example of the use of the binary tree in software filtering shown in Figure 3, a section of the packet stream which has been passed along

by the hardware PES filters to the software filter is checked against a root node 300A to determine if this section is a section which is to be accepted.

If the analyzed section does not match the criteria recorded in the level one node 300A, then the analyzed section is compared with the criteria stored in the level one node 300B. If the analyzed section does not match any of the criteria's stored in any of the level one nodes 300A – 300D, then the section is rejected. If, however, the analyzed section matches one of the criteria's associated with one of the level one nodes 300A – 300D, then the analyzed section is compared to a level two set of criterias associated with the level one criteria which matched with the analyzed section. For example, if an analyzed section did not match a criteria associated with level one node 300A, but did match with a criteria associated with level one node 300B, then the analyzed section would then be compared to a set of level two nodes 300F – 300H associated with the matching level one node 300B.

If the analyzed section does not match with any of the level two to nodes 300F – 300H associated with the matching level one node 300B, then the analyzed section is rejected. If, however, the analyzed section matches a criteria associated with a level two node, such as node 300H, then the analyzed section is compared with criterias associated with a set of level three nodes 300K – 300N.

If there are only two levels of criteria associated with a particular table, such as the NIT and BAT in the example described above, then the analyzed section would be accepted after matching the level two criteria of node 300H.

This process continues until the analyzed section is either rejected by failing to match any of the criterias associated with a node at a particular level, or the analyzed section is accepted after being matched with all criterias stated at all levels. The leaves 302A – 302F are the final elements of a particular branch of the binary tree, wherein the analyzed section is accepted if it

reaches a leaf 302A – 302F. For example, an analyzed section may progress through a branch 300A – 300B, 300F – 300H, 300K – 300M- 300L, to a final leaf 302E.

A predetermined number of nodes and leaves are preallocated in memory to accommodate the software filtering process. An example of the memory preallocated for nodes and leaves may be approximately 27 kilo bytes for a given number of nodes (such as 930 nodes) and a given number of leafs (such as 250 leaves). This preallocated amount of memory is typically a substantial portion of memory in the set top box of an interactive television system since set top box typically only has a small amount of memory, such as approximately ½ mega byte to one mega byte.

Once a set of data has been accepted by the software filter, it is typically desirable to determine which sections associated with a particular system information table has been properly received. Figure 4 illustrates a conventional method for keeping track of whether a particular section associated with a given system information table has been received. An array of 32 bites is typically reserved for keeping track of whether a section of a particular system information table has been received. This 32-bit array is commonly referred to as a section map. A system information table may include a large number of sections, for example 256 sections. Each bit within the 32 bytes of the section map may indicate whether a specific section number has been received. In the example shown in Figure 4, byte 1 includes bits 1 to 8 which may indicate whether section numbers 0 – 7 have been received, the second byte also includes 8 bits which may include whether section numbers 8 to 15 have been received, and so on, up to byte 32 indicating up to whether section 248 to 255 has been received.

A table may include 256 sections. Accordingly, enough memory space is typically reserved to be able to indicate the receipt of 256 sections. However, only two or three sections are usually included in a system information table, and most of the memory space reserved for the section map typically remains unused.

Although this process may be effective as a software filtering mechanism, it would be desirable to reduce the memory requirements of the software filtering process without decreasing the speed of the software filter process. Additionally, it would also be desirable to have a bit map system which requires less memory. The present invention addresses such needs.

Figure 5 is a data structure according to an embodiment of the present invention. The data structure 400 shown in Figure 5 include multiple fields, herein shown in rows 406 – 412 and columns 402 – 404. In this example, it is assumed that there may be up to four levels of criterias that the software filter may search for a match to a received table data. In column 402, the requested criteria is listed in a field associated with each level 406 – 412. In column 404, an index to the next subtable node to be used if a match is found with an associated requested criteria 402. For each field in the requested criteria column 402 and the index column 404, a predetermined number of bytes, such as 2 bytes, may be reserved.

The subtable node 400 is also shown to include a level field 414. An example of memory space which may be reserved for the level field 414 is one byte. The level field may include a value indicating a level, in this example 1 – 4, which indicates which levels are valid fields. For example, if level field 414 indicated level one, then only level one (row 406) needs to be checked for a match for a given section. In another example, if the level field indicates 3, then levels one (row 406), two (row 408), and three (row 410) are valid and level four (row 412) does not need to be checked for a match with a received section.

Area 416 may be reserved for additional information. For example, area 416 may be several fields, such as seven fields, used to store information, such as system information table version number, flags for indicating whether the subtable node 400 is still valid, a counter counting how many sections are waiting to be processed, a client index, a subtable filter index, a tag, and a section map. Further details of the section map will later be discussed in conjunction with Figures 9A – 11. A client index may be a database structure pointing to a root subtable node

(first subtable node to be analyzed) associated with this subtable node 400. A subtable filter index may be an index to other structures within the software filter, not related to this subtable node binary tree. The tag may be a tag originating from a client for use for various purposes determined by the client.

Figure 6 shows an example of a set of subtable nodes 400A – 400D according to an embodiment of the present invention. A section is received from the hardware PES filter by the software filter according to an embodiment of the present invention. This section is first analyzed through a root subtable node 400A. A root subtable node 400A is the first subtable node to be compared with a received section. If there are two suitable subtable nodes which have a criteria 402 at level one 406 which matches the analyzed section, then the subtable node with the highest number in the level field 414 is evaluated before the subtable node with the lower number in the level field 414. Accordingly, subtable nodes with criterias at level one which match the section are evaluated sequentially by the highest level field 414 down to the lowest level field 414.

In an embodiment of the present invention, if a field of a subtable node 400 is valid, then the previous fields may also be assumed valid within the same subtable node 400. The previous fields are the fields in column 402 which are in lower levels 406 – 412 then the field which is valid. For example, if level field 414 identifies level three (row 410), then it is assumed that levels one (row 406), and two (row 408) are also valid.

In the example shown in Figure 6, a section is evaluated in the first level 406A of column 402A of a root node 400A. If there is a match between the section and the criteria associated with the first level 406A of the root node 400A, then the second level 408A of column 402A of the root node 400A is checked for a match, and if a match is found at the second level 408A of column 402A, then the third level 410A of column 402A is checked, and so on until all the levels 406A – 412A of column 402A have been checked for a match in the root node 400A. If a match

is not found at a level 406A-412A in the root node 400A, then an associated index located in the same level in column 404A of the root node 400A is referenced to determine which subtable node to check next. For example, if a match has not occurred in level one 406A of column 402A of the root node 400A, then rather than checking level two 408A of the root node 400A, an index located in the field of row 406A, column 404A, is checked to determine the next subtable node to be referenced. In this example, subtable node 400B is the table indicated in the index associated with the matching criteria of level one of the root node 400A. In table 400B, the same level that failed to find the match in the root node 400A is checked for a match. In our example, since there was no match found in level one of the root node 400A, level one 406B of the next subtable node 400B is then checked. If the first level 406A of the root node 400A is a match with the section, then the section is checked with the second level 408A of column 402A of root node 400A to check for a match.

In the next table 400B, if the first level to be checked (which is the corresponding level to the level which failed to match with the section in the previous subtable node) also fails to match, then the index 404B corresponding to the level which failed to match is checked to determine the next subtable node to be analyzed. In this example, the first level to be analyzed in subtable node 400B is level one 406B of column 402B since the corresponding level in the root node 400A which failed to match was also in level one 406A.

If a match is found in level one 406B of column 402B of the subtable node 400B, then the next level 408B of the subtable node 400B is checked for a match. If, however, there is no match found in the first level 406B of column 402B of subtable node 400B, then the associated index located in the field of row 406B, column 404B, is referenced to determine the next subtable node 400D to be checked. If a match was found in level one 406B of column 402B of subtable 400B, then the next level, level two 408B of subtable node 400B is checked for a match. If a match

does not occur at that level, then the associated index located in a field in row 408B, column 404B, is referenced to determine the next subtable node 400C to be checked.

Assume a match has not been found in level two 408B of subtable node 400B. The associated index in column 404B directs the next comparison to be with level two 408C of subtable node 400C.

Figures 7 and 8 are flow diagrams of a method according to an embodiment of the present invention for filtering sections from a packet stream. A level is set to one, and a subtable node (SN) is set equal to root (step 500). A field in the section corresponding to the level (level ID) is retrieved from a section to be analyzed (step 502). Examples of the level's ID include criterias such as table ID, network ID, bouquet ID, original network ID, transport stream ID and service ID. The level's ID may be found in the section header.

Figures 12A – 12D are illustrations of examples of various section headers. Figure 12A shows an example of a NIT header 900a which includes fields 902a – 902d. In this example, field 902a is a table ID, field 902b is a network ID, field 902c is a version number, and field 902d includes a current section number and the last section number. Figure 12B shows an example of a BAT header 900b with fields 902a' – 902d'. In this example, field 902a' is a table ID, field 902b' is a bouquet ID, field 902c' is a version number, and field 902d' includes a current section number as well as the last section number. Figure 12C shows an example of an STD header 900c with fields 902a" – 902e. In this example, field 902a: is a table ID, field 902b: is a transport stream ID, field 902c" is a version number, field 902d" includes the current section number and the last section number, and field 902e is an original network ID. Figure 12D shows an example of an EIT header 900d with fields 902a'" – 902f. In this example, field 902a'" is a table ID, field 902b'" is a service ID, field 902c'" is a version number, field 902d'" includes a current section number and a last section number, field 902e' is a transport stream ID, and field 902f is an original network ID.

In the examples shown in Figures 12A – 12D, fields 902a – 902a''' may be level one IDs, fields 902b – 902b''' may be level two IDs, fields 902e – 902e' may be level three IDs, and field 902f may be a level four ID. For example, these fields include level IDs that may be retrieved in step 502 of Figure 7. Note that the number of valid level fields varies depending on the section type (NIT, BAT, SDT etc.). Fields 902a, 902a', 902a'', and 902a''' are shown to include table IDs. The contents of these fields 902a – 902a''' may determine the contents of the remainder level IDs 902b, 902b', 902b'', 902e, 902b''', 902e', and 902g.

In addition to the level IDs, the header 900 may include various other information, such as a version number, and a current section number as well as the last section number. Further details of fields that may be included in section headers may be found in Digital Video Broadcasting (DVB) Standard EN 300468.

It is then determined whether the level ID (retrieved value of the section's field) retrieved from the section matches the subtable node's criteria in the same level (step 504 of Figure 7). For example, if the level is set to one, and the subtable node is set to the root node, such as root node 400A of Figure 6, and the criteria associated with the field in level one 406A of column 402A of root node 400A of Figure 6 is a particular table ID, then the level one identification found in the section header of the section being analyzed is compared to the table ID identified in level one of the root node 400A to see if they match.

If the level ID retrieved from the section is not the same as the criteria identified in the subtable node of the current level, then the next subtable node is determined (step 506). The next subtable node may be determined by the index in column 404 of the current level row of the subtable node, as shown in Figure 6.

It is then determined whether this subtable node is valid (step 508). A flag indicating whether a subtable node is valid may be stored in area 416 of the node 400, as shown in Figure 5. If the subtable node is not valid, then the current section is rejected (step 510). If, however, the

current subtable node is valid, then it is again determined whether the section's level ID is the same as the criteria in the current subtable node at the same level (step 504).

If the level ID of the section is the same as the subtable node's criteria for the same level (step 504), then it is determined whether the level is greater or equal to the current subtable node's level field (step 512), such as level field 414 of Figure 5. If the level is less than the subtable node's level field, then the next level is considered (step 514) (set level = level + 1). A field in this section corresponding to this next level (next level ID) is then retrieved (step 502).

If the level is greater than or equal to the level field of the subtable node (step 512), such as the level field 414 of Figure 5, then it is determined whether a version in the current subtable node is equal to a version in the section (step 550 of Figure 8). The version number may be located in the additional field 416 of the subtable node 400 shown in Figure 5. This version number stored in the subtable node may be compared to a version number stored in the section header, such as in field 902e of the header 900 of Figure 12. If the version number in the subtable node is not equal to the version number in the section header, then the version in the subtable node is updated, the section map is cleared and a new section map is created (step 552). When a section with a new version is received, all of the sections associated with the new version should be reacquired to ensure that all changes have been received. Accordingly, the section map that has been tracking the receipt of the old version may be cleared, and a new section map may be created to keep track of the receipt of sections in the new version. Details of manipulations of the section map is later discussed in conjunction with Figures 9A – 11. Thereafter, the section map is updated with the sections in the new version as they are acquired (step 554). The section is then accepted (step 556).

If the version in the subtable node is equal to the version in the section (step 550), then it is determined whether this section has been received before (step 558). It is possible that portions of a version of a table has been received with some missing sections. Accordingly, a

check is performed to determine whether this section is one of those missing sections. Details of this determination are later discussed in conjunction with Figure 10. If this section has been received before, then the section is rejected (step 560). If, however, this section has not been received before, then the section map is updated to indicate that this section is being accepted (step 554) and the section is accepted (step 556).

Figures 9A – 9B illustrate a section map according to an embodiment of the present invention. A section map may be located in the area 416 of the subtable node shown in Figure 5. The section map includes a field with multiple bits, for example a 32-bit field with the last 16 bits 602 indicating whether the first 16 sections have been received. Each bit in the 16-bit area 602 may indicate whether a corresponding section has been received. Another bit 604 may be used to indicate the function of the first 15 bits 606. For example, when the indicator bit 604 is set to one, the first set of bits 606 may indicate the received status of whether the next 15 sections have been received. If, however, the indicator bit 604 is zero, then it may indicate that the first section 606 may be an index to point to the next section map field.

Figure 9B shows an example of the indicator bit indicating an index when it is set to zero and the status of the next 15 sections when the indicator bit is set to one. In this example, a first 32-bit field 600A has collected the status of the first 16 sections in a second portion 602A. In this example, the second portion 602A includes 16 bits, each bit indicating whether a section has been received. The indicator bit is located between the first portion 602A and a second portion 606A. In this example, the indicator bit 604A is set at zero and is one bit. The zero in the indicator bit 604A indicates that the first portion 606A is an index pointing to the next section map 600B. The next section map 600B may also be a 32-bit field with a first portion 606B, a second portion 602B, and an indicator bit located between the first portion 606B and the second portion 602B. The second portion 602B records the status of the second set of 16 sections. In this example, the indicator bit 604B is also set to zero, thus indicating that the first portion 606B

of the second section map 600B is also an index pointing to a third section map 600C. The third section map 600C also stores a third set of bits indicating the status of 16 sections. However, the indicator bit 604C is set to one, indicating that this section map 600C is the last section map and that the first portion 606C indicates the status of the next set of sections, in this example the next 15 sections.

Figure 10 is a flow diagram of a method according to an embodiment of the present invention for determining whether a section has already been received, such as the determination of step 558 of Figure 8. A section number is retrieved from the section (step 700). A first section map, such as section map 600A of Figure 9B, is analyzed (step 702). It is then determined whether the section number is less than the number of bits preallocated for a section map field (step 704). In this example, it is determined whether the section number is less than 31 (note that sections are counted from 0 to 255 in this example).

If the section number is not less than the number of bits preallocated for a section map field, then the section number is decremented by the number of bits preallocated for the second portion of the section map field (step 706). In this example, the section number is decremented by 16. The first portion of the current section map is used to point to the next section map (the next 32 bit section map field) (step 708). Thereafter, it is again determined whether this section number is less than the number of bits in the section map field (step 704).

If the section number is less than the number of bits in the section map field (step 704), then it is determined whether the section number is less than the number of bits in the second portion (step 716). In this example, the number of bits in the second portion is fifteen. If the section number is not less than the number of bits in the second portion, then it is determined whether the current section map is pointing to another section map (step 718). An example of how this determination can be made is to set the indicator bit to indicate pointing to another section map. If this section map is pointing to another section map, then the section number is

decremented by the number of bits in the second portion (step 706). If, however, this section map is not pointing to another section map (step 718), or if the section number is less than the number of bits in the second portion (step 716), then it is determined whether a bit associated with the current section indicates whether the section has been received (step 710). For example, a one may be used in the bit associated with the current section to indicate that it has been received (step 714). If the associated bit is not set in the pointed section map (step 710), then the section has not yet been received (step 712).

Figure 11 is a flow diagram of a method according to an embodiment of the present invention for creating a new section map, such as in step 552 of Figure 8. The last section number from the current section is retrieved (step 800). Each section identifies its own section number, as well as the last section number so that the receiving system knows how many more sections are still yet to come. A first section map, such as section map 600A of Figure 9B, is used (step 802). It is then determined whether the last section number is less than the number of bits in the section map field (step 804). In this example, it is determined whether the last section number is less than 31. If the last section number is less then the number of bits in the section map field (step 804), then the indicator bit is set to one to indicate that there are no further section maps (step 806). In this example, the indicator bit is bit 15. Accordingly, the creation of the new section map is complete.

If the last section number is not less than the number of bits in the section map field (step 804), then the last section number is decremented by the number of bits in the second portion of the section map field (step 808). An example of the number of bits of the last section of the section map field is the 16 bits located in the second portion 602 of Figure 9A.

The indicator bit is then set to zero to indicate that there are further section maps (step 810). A new section map is then retrieved from the system, such as from random access memory (RAM) and linked to the current section map (step 812). The new section map is then pointed to

by the first portion, such as first portion 606 of Figure 9A, of the previous section map (step 814). It is then again determined whether the last section number is less than the number of bits in the section map field (step 804), and so on, until the process is complete.

A method and system for information filtering has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor.

Although the present invention has been described in accordance with the embodiment shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiment and these variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.

CLAIMS

What is claimed is:

1. A method for filtering information in an interactive television environment, the method comprising:

providing a data structure, wherein the data structure includes a plurality of levels;

comparing information with a first criteria, wherein the first criteria is related to a first one of the plurality of levels; and

comparing the information with a second criteria, wherein the second criteria is related to a second one of the plurality of levels, this comparison occurring if the information matches the first criteria.

2. The method of claim 1, further including comparing the information with a third criteria, wherein the third criteria is related to a second data structure, this comparison occurring if the information does not match the first criteria.

3. The method of claim 2, wherein the third criteria is also related to a third level, wherein the third level is related to the first one of the plurality of levels.

4. The method of claim 1, further including receiving the information.

5. The method of claim 1, wherein the information is a section.

6. The method of claim 1, wherein the data structure also includes a field indicating which one of the plurality of levels is valid.

7. The method of claim 1, wherein the data structure also includes a section map.

8. The method of claim 1, wherein the data structure also includes an index to a second data structure.

9. The method of claim 8, wherein the index is used if the information does not match the first criteria.

10. The method of claim 1, wherein the information is a section header.

11. A system for filtering information in an interactive television environment, the system comprising:

means for providing a data structure, wherein the data structure includes a plurality of levels;

means for comparing information with a first criteria, wherein the first criteria is related to a first one of the plurality of levels; and

means for comparing the information with a second criteria, wherein the second criteria is related to a second one of the plurality of levels, this comparison occurring if the information matches the first criteria.

12. A system for filtering information in an interactive television environment, the system comprising:

a memory configured to provide a data structure, wherein the data structure includes a plurality of levels; and

a processor coupled to the memory, the processor being configured to compare the information with a first criteria, wherein the first criteria is related to a first one of the plurality of levels; and the processor also being configured to compare the information with a second criteria, wherein the second criteria is related to a second one of the plurality of levels, this comparison occurring if the information matches the first criteria.

13. The system of claim 12, further including comparing the information with a third criteria, wherein the third criteria is related to a second data structure, this comparison occurring if the information does not match the first criteria.

14. The system of claim 13, wherein the third criteria is also related to a third level, wherein the third level is related to the first one of the plurality of levels.

15. The system of claim 12, further comprising a front end coupled to the processor, the front end being configured to receive the information.

16. The system of claim 12, wherein the data structure also includes a section map.

17. The system of claim 12, wherein the data structure also includes an index to a second data structure.

18. The system of claim 17, wherein the index is used if the information does not match the first criteria.

19. A computer program product for filtering information in an interactive television environment, comprising:

computer code for receiving information;

computer code for providing a data structure, wherein the data structure includes a plurality of levels;

computer code for comparing the information with a first criteria, wherein the first criteria is related to a first one of the plurality of levels;

computer code for comparing the information with a second criteria, wherein the second criteria is related to a second one of the plurality of levels, this comparison occurring if the information matches the first criteria; and

a computer readable medium that stores the computer codes.

20. The computer program product of claim 19, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

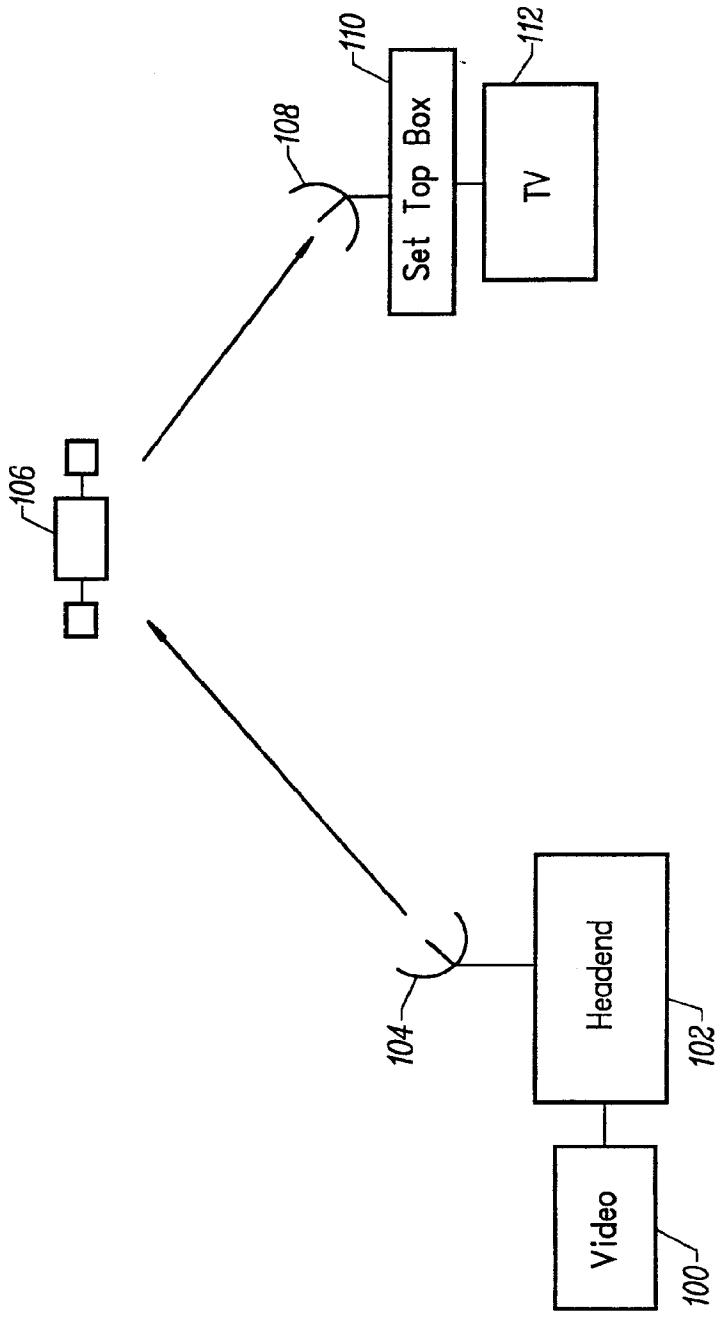


FIG. 1

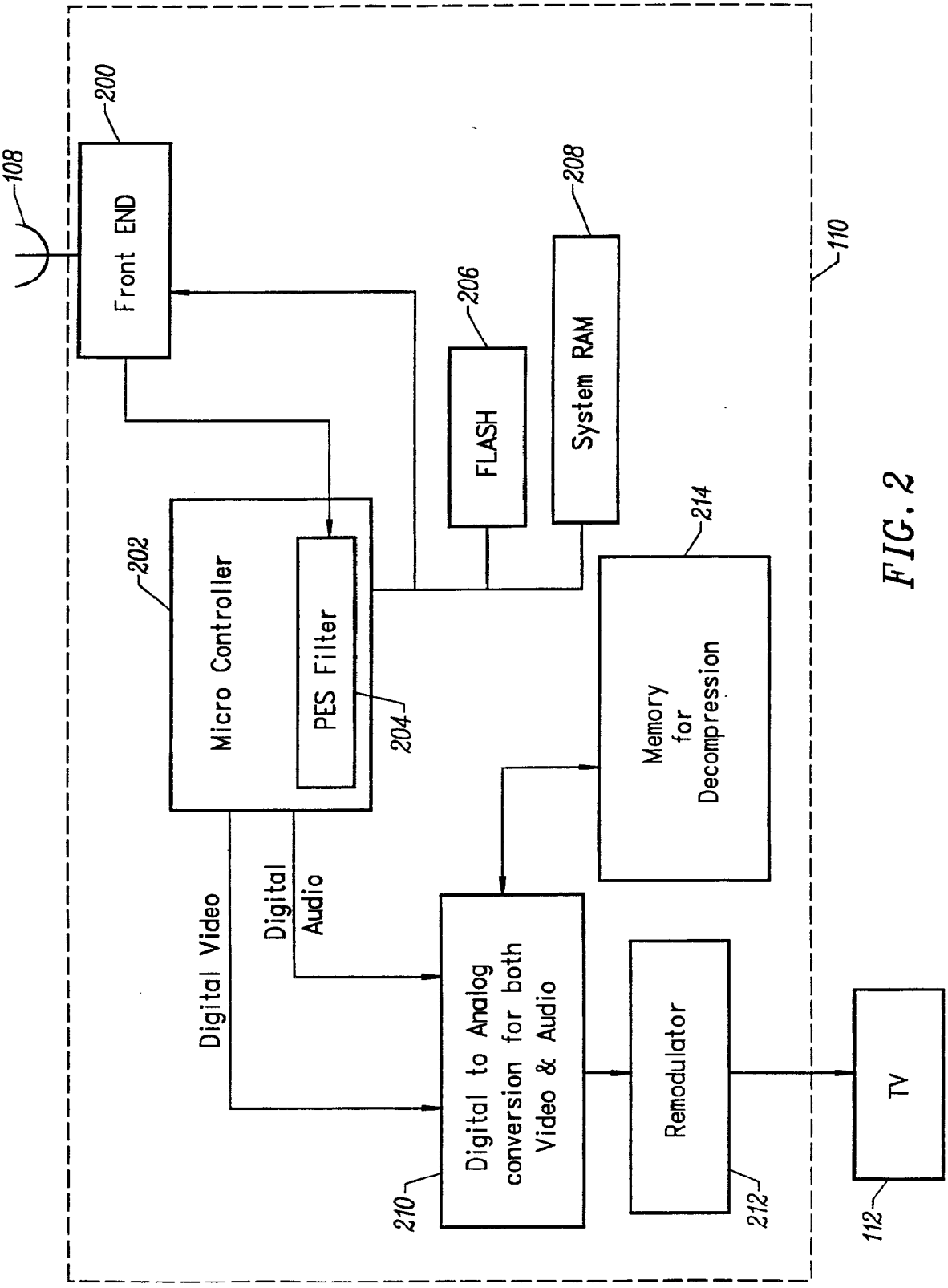

108
200
Front END
208
System RAM
110
206
FLASH
202
Micro Controller
PES Filter
204
214
Memory for Decompression
Digital Video
Digital Audio
Digital to Analog conversion for both Video & Audio
Remodulator
TV
210
212
112

FIG. 2

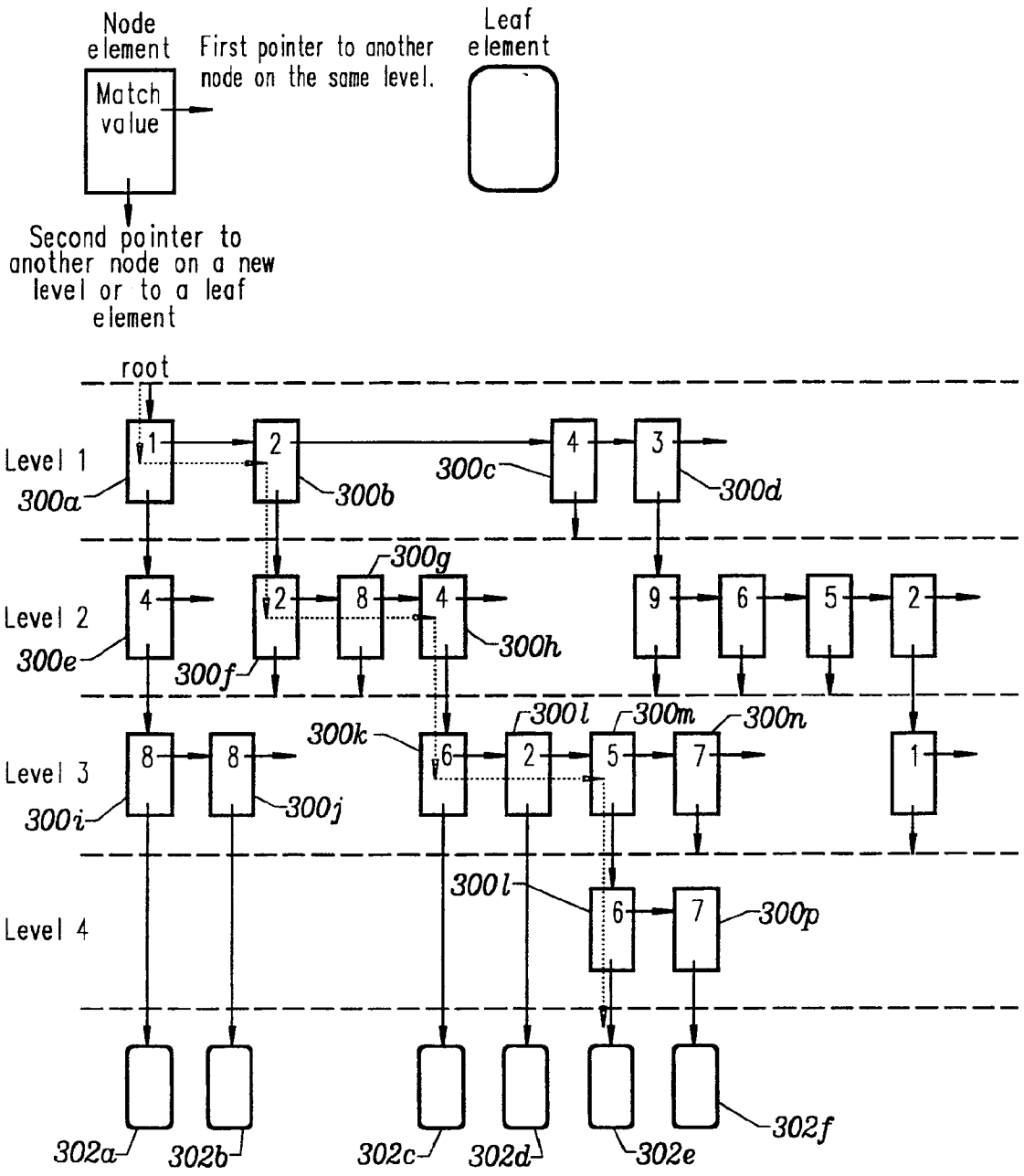

Node element
Match value
First pointer to another node on the same level.
Leaf element
Second pointer to another node on a new level or to a leaf element
root
Level 1
Level 2
Level 3
Level 4
300a
300b
300c
300d
300e
300f
300g
300h
300i
300j
300k
300l
300m
300n
300p
302a
302b
302c
302d
302e
302f

FIG. 3

Byte	Bit	Section #
1	1	0
	.	.
	.	.
	.	.
	8	7
2	1	8
	.	.
	.	.
	.	.
	8	15
.	.	.
.	.	.
.	.	.
32	1	.
	.	.
	.	.
	.	.
	8	255

FIG. 4

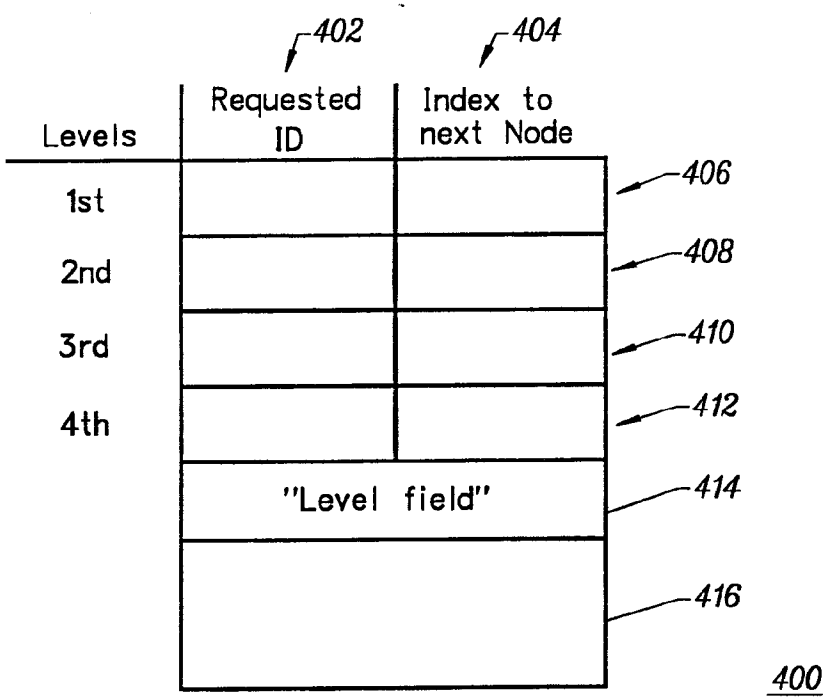

402
404
Levels
Requested ID
Index to next Node
1st
2nd
3rd
4th
406
408
410
412
"Level field"
414
416
400

FIG. 5

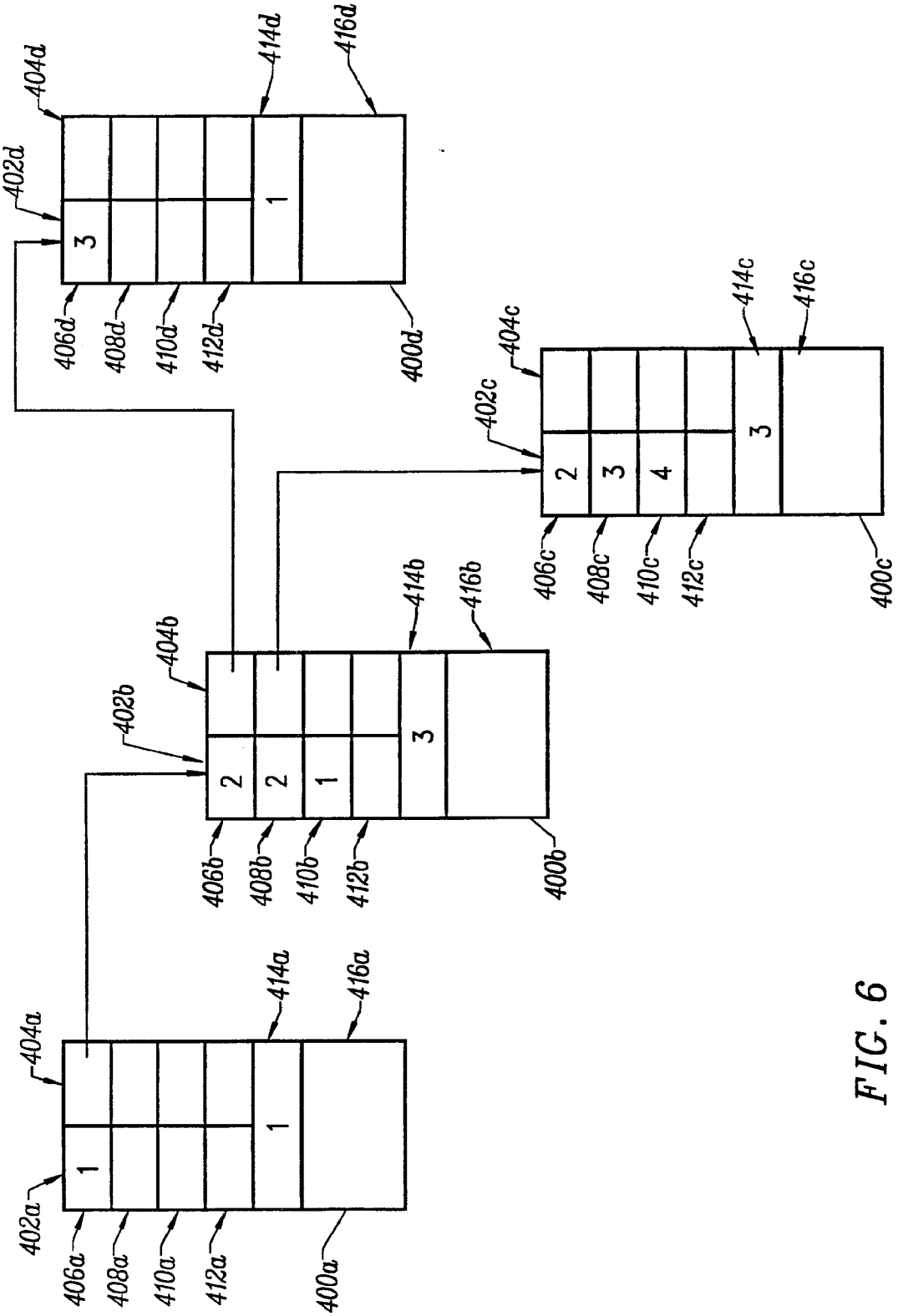

404d
414d
416d
402d
3
1
406d
408d
410d
412d
400d
404c
414c
416c
402c
2
3
4
3
406c
408c
410c
412c
400c
404b
414b
416b
402b
2
2
1
3
406b
408b
410b
412b
400b
404a
414a
416a
402a
1
1
406a
408a
410a
412a
400a

FIG. 6

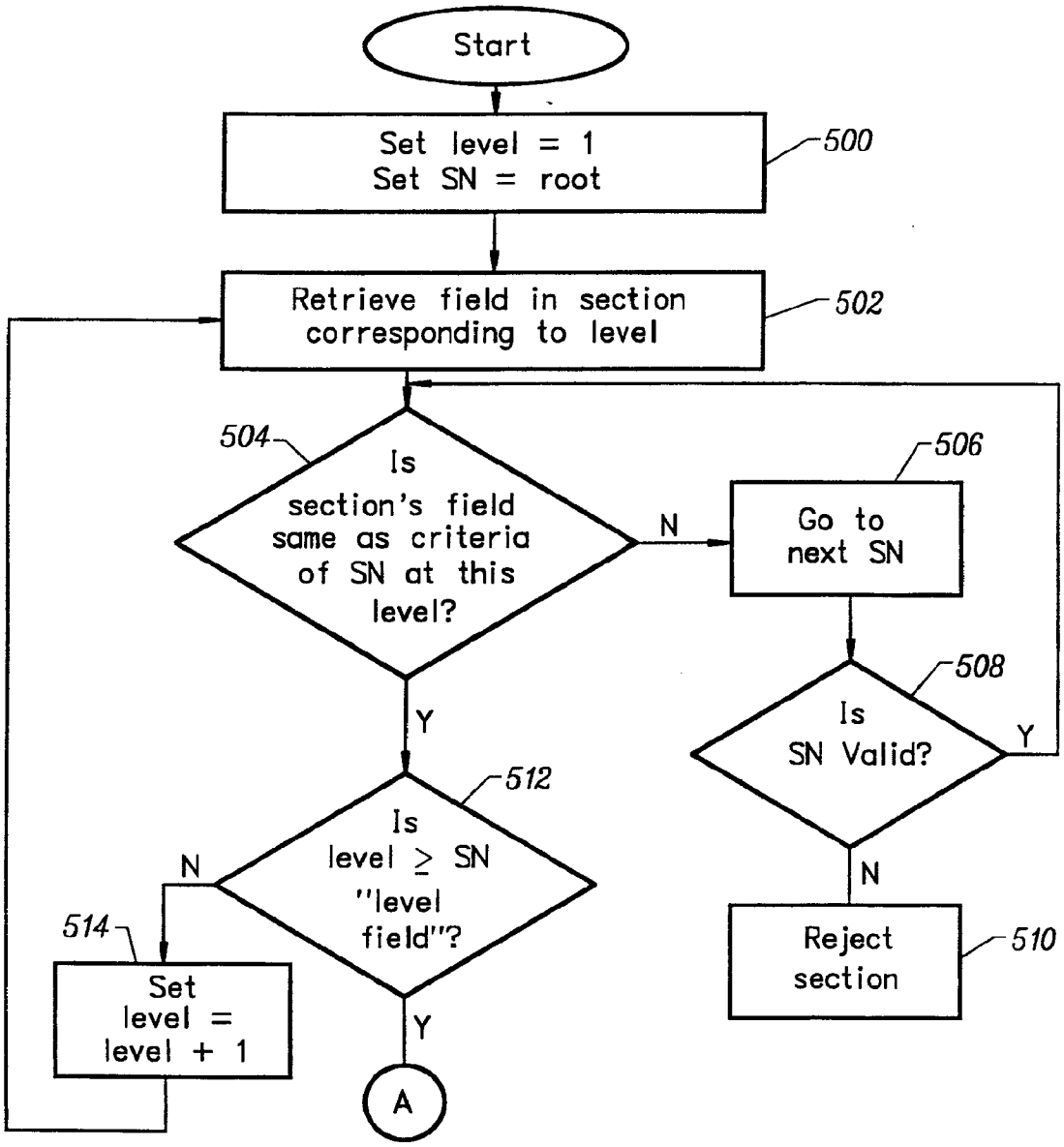

Start
Set level = 1
Set SN = root
500
Retrieve field in section corresponding to level
502
504
Is section's field same as criteria of SN at this level?
N
506
Go to next SN
508
Is SN Valid?
Y
N
Reject section
510
Y
512
Is level ≥ SN "level field"?
N
514
Set level = level + 1
Y
A

FIG. 7

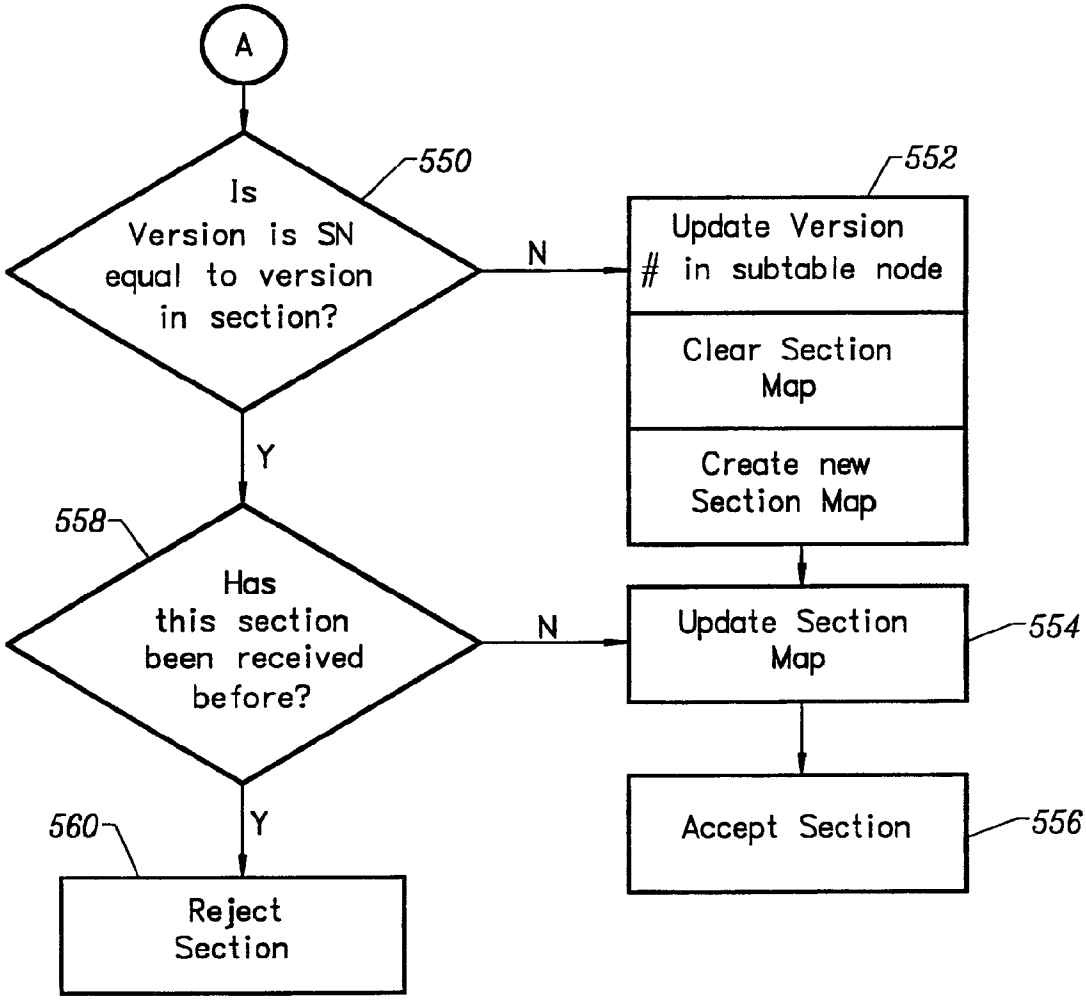

A
550
Is Version is SN equal to version in section?
N
552
Update Version # in subtable node
Clear Section Map
Create new Section Map
Y
558
Has this section been received before?
N
Update Section Map
554
Accept Section
556
Y
560
Reject Section

FIG. 8

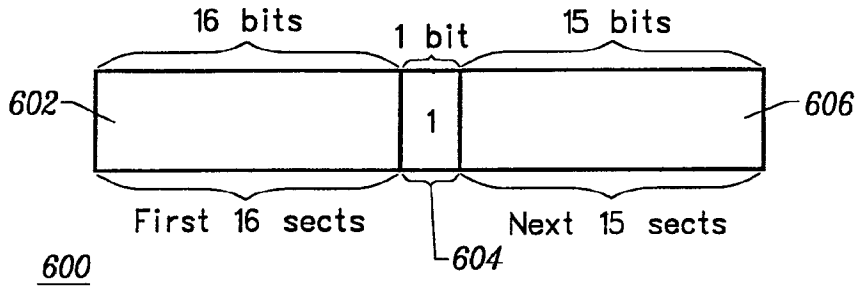

16 bits
1 bit
15 bits
602
1
606
First 16 sects
Next 15 sects
604
600

FIG. 9A

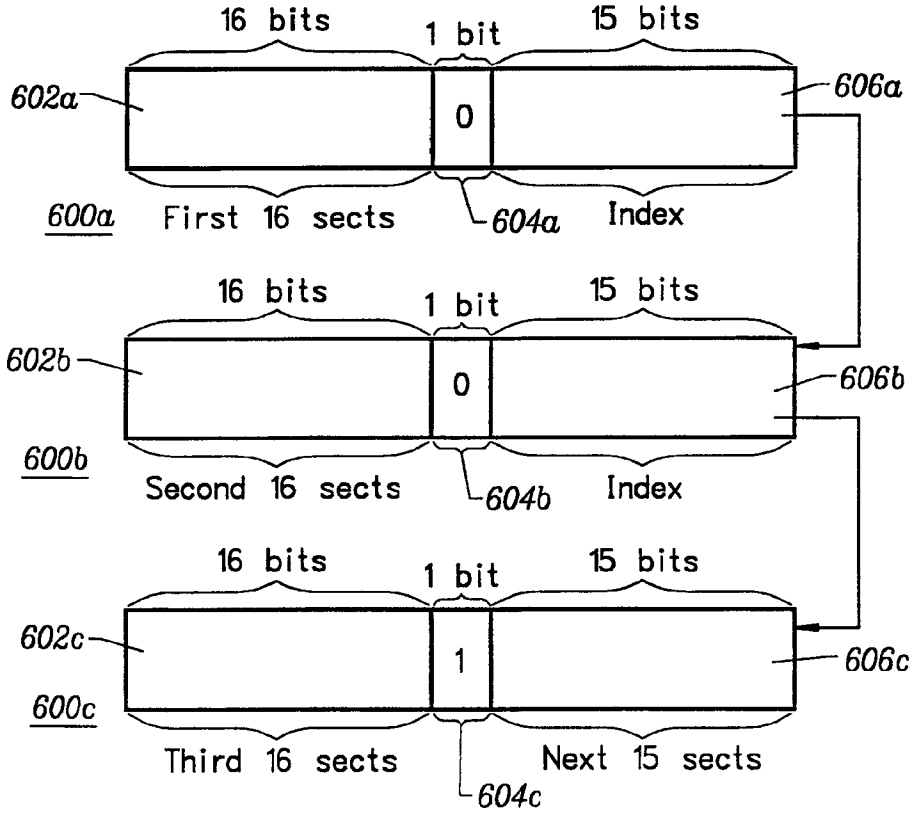

16 bits
1 bit
15 bits
602a
0
606a
600a
First 16 sects
604a
Index
16 bits
1 bit
15 bits
602b
0
606b
600b
Second 16 sects
604b
Index
16 bits
1 bit
15 bits
602c
1
606c
600c
Third 16 sects
Next 15 sects
604c

FIG. 9B

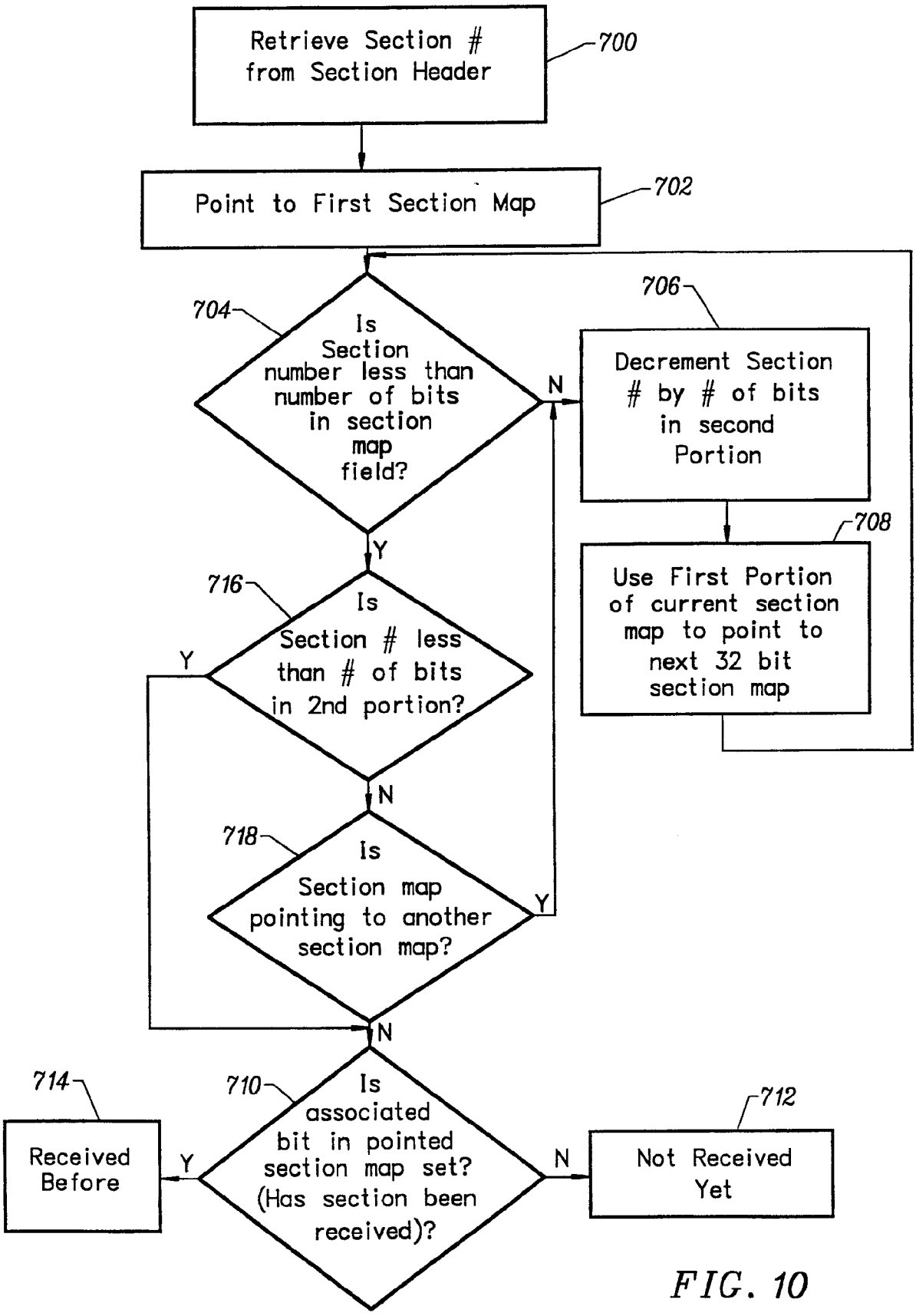

Retrieve Section # from Section Header
700
Point to First Section Map
702
704
Is Section number less than number of bits in section map field?
N
706
Decrement Section # by # of bits in second Portion
708
Use First Portion of current section map to point to next 32 bit section map
Y
716
Is Section # less than # of bits in 2nd portion?
Y
N
718
Is Section map pointing to another section map?
Y
N
714
Received Before
710
Is associated bit in pointed section map set? (Has section been received)?
Y
N
712
Not Received Yet

FIG. 10

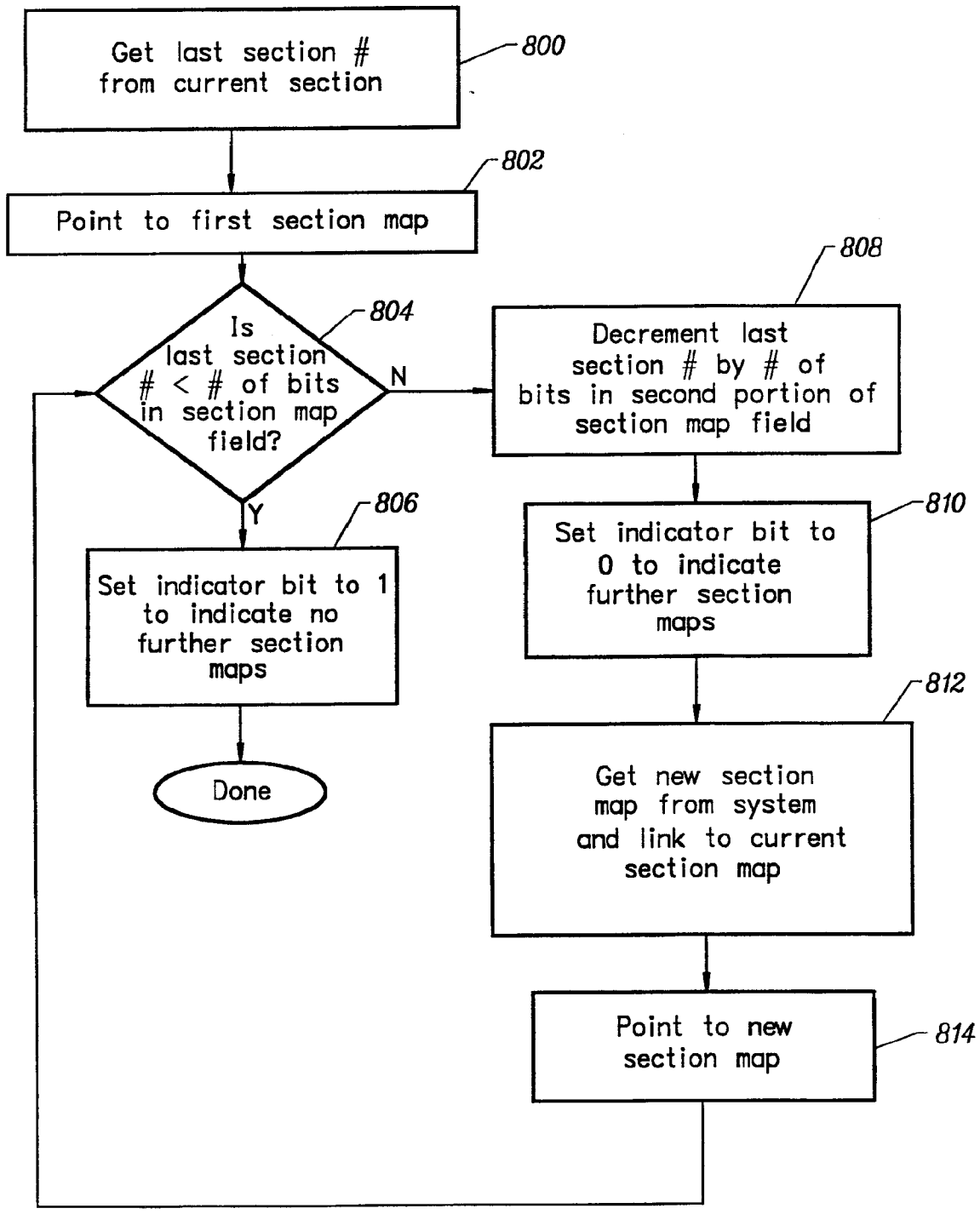

Get last section # from current section
800
Point to first section map
802
Is last section # < # of bits in section map field?
804
N
Y
Decrement last section # by # of bits in second portion of section map field
808
Set indicator bit to 1 to indicate no further section maps
806
Set indicator bit to 0 to indicate further section maps
810
Done
Get new section map from system and link to current section map
812
Point to new section map
814

FIG. 11

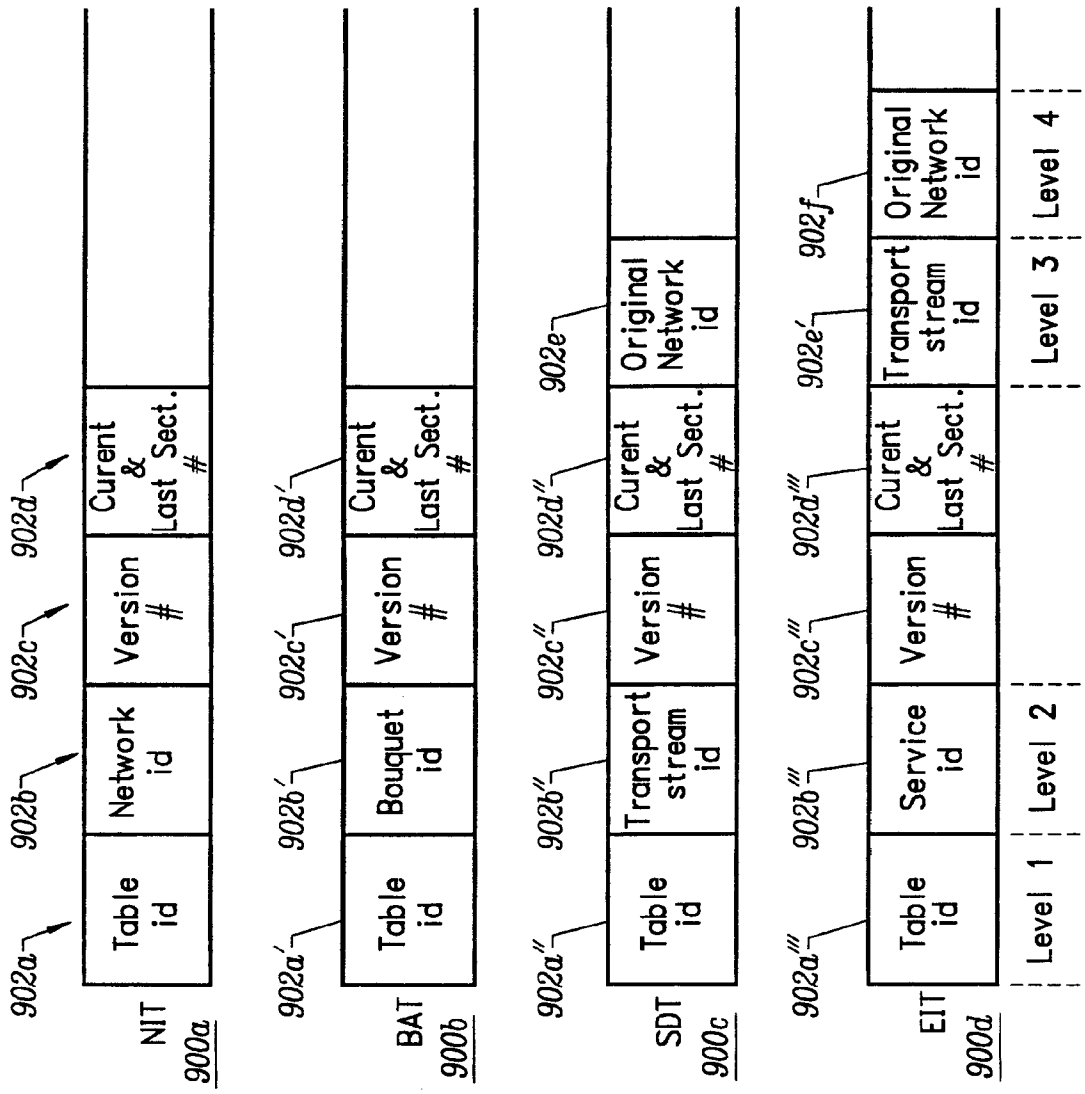


FIG. 12A

FIG. 12B

FIG. 12C

FIG. 12D

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/25730

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :Please See Extra Sheet.
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/327; 348/6, 10, 460, 461, 473, 906; 455/3.1, 4.1, 4.2, 6.1, 6.2

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,600,378 A (WASILEWSKI) 04 February 1997, see whole document.	1-20
X, P	US 5,864,358 A (SUZUKI et al) 26 January 1999, see whole document.	1-20
X, P	US 5,909,257 A (OHISHI et al) 01 June 1999, see whole document.	1-20
X, P	US 5,943,605 A (KOEPELE, JR.) 24 August 1999, see whole document.	1-20

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search
04 JANUARY 2000

Date of mailing of the international search report
02FEB2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
JOHN W. MILLER
Telephone No. (703) 308-0000

Form PCT/ISA/210 (second sheet)(July 1992)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/25730

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A, E	US 5,999,216 A (KAARS) 07 December 1999, see whole document.	1-20

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/25730

A. CLASSIFICATION OF SUBJECT MATTER:
IPC (6):

H04N 7/10, 7/08

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

345/327; 348/6, 10, 461, 473, 906; 455/6.2